

Projeto do Banco de Dados Amazon GO (v2.0)

Eduardo Henrique Souza Paraíso

Dezembro 2, 2020

RESUMO

O presente relatório apresenta o projeto do banco de dados Amazon GO. Especificamente, o relatório apresenta o resultado das atividades de especificação do minimundo, análise de requisitos, projeto conceitual, projeto lógico e projeto físico do Amazon GO (v2.0). As informações necessárias para a realização das atividades de modelagem foram coletadas a partir de especificações obtidas através de conteúdo disponibilizado na internet pela proprietária e vídeos de experiências de usuários. O banco de dados AMAZON GO (v2.0) foi concebido dentro do paradigma relacional utilizando como base o modelo relacional, sendo constituído por um conjunto de seis entidades, seis relacionamentos e uma média de seis atributos por entidade. Além disso, as entidades e relacionamentos conceituais deram origem a um esquema de implementação composto por oito tabelas e oito restrições de integridade referencial entre elas. Atributos com alta taxa de acesso tais como CPF do cliente, CNPJ da loja filiada, localização de um determinado produto foram indexados para uma melhor eficiência na realização de transações pelo banco de dados do projeto Amazon GO, assim como chaves estrangeiras existentes também foram indexadas.

1 Introdução

Com base em Seattle, EUA, a loja usa a tecnologia avançada de compras da Amazon, desenvolvida para reduzir as filas nos caixas. O supermercado futurista, testado pela gigante da web é repleto de centenas de câmeras e dispositivos de rastreamento, para que os clientes possam passear pelos corredores enquanto estão conectados a um aplicativo, pegar os produtos que desejam e simplesmente sair. A escolha da abordagem de banco de dados utilizada para organização e manutenção desses registros influencia a efetividade com que as transações são efetuadas. O objetivo do presente trabalho é propor um projeto de banco de dados para um sistema de um novo sistema de lojas de varejo, que poça futuramente ser aberto a novas concepções similares que utilizem inteligência artificial e visão computacional para mudar a perspectiva de compras de seus clientes. Particularmente, propõe-se uma especificação de minimundo, análise de requisitos, projeto conceitual, projeto lógico e projeto físico do banco de dados Amazon GO, que em sua versão 1.0 é concebido no paradigma relacional, utilizando como base o modelo relacional e podendo ser implementado em sistemas gerenciadores de banco de dados (SGBD) relacionais comerciais.

A Seção 2 apresenta a especificação de minimundo do banco de dados, incluindo uma descrição textual das principais características e restrições de dados. A Seção 3 apresenta o projeto conceitual do banco de dados, incluindo diagrama entidade-relacionamento (ER) que representa de forma gráfica as principais definições conceituais, próximas à forma como os usuários percebem o banco de dados. A Seção 4 apresenta o projeto lógico do banco de dados, incluindo diagrama relacional que representa de forma gráfica as principais definições para implementação do banco de dados dentro do paradigma relacional. A seção 5 apresenta as conclusões obtidas na construção deste modelo. A Seção 6 apresenta o código SQL devidamente separados entre a criação das tabelas, indexação, restrições e povoamento do banco com dados artificiais. Neste trabalho não foi utilizado triggers, devendo este tipo de restrição ser implementada em versões futuras do projeto.

2 Especificação do Minimundo

Essa seção apresenta a descrição textual de minimundo da Amazon GO (v2.0), um banco de dados para um sistema inovador de varejo que possibilita o cliente comprar produtos em uma loja física sem a necessidade de enfrentar filas ou passar por caixas eletrônicos. A Amazon GO é composta por uma rede de franquias, franquias estas compostas por um código identificador da franquia, o CNPJ do franquiado e o endereço do empreendimento. Estas franquias possuem funcionários - compreendidos por uma identificação de funcionário, cargo que exerce, nome, CPF e o identificador da franquia na qual faz parte - que atuam apenas na reposição de produtos do estoque nas prateleiras e na resolução de possíveis dúvidas ou problemas do cliente, porém cada funcionário trabalha em sua franquia específica. Todas as franquias são compostas por diversos produtos - com: identificação do tipo de produto, localização (setor e prateleira), nome, marca, data de validade, lote de fabricação e preço unitário em reais (BRL) - e cada produto não é exclusivo de uma única filial. O cliente deve possuir cadastro válido - informando: CPF, nome, endereço, número

válido de celular, e-mail, cartões de créditos válidos e data de nascimento - em uma conta Amazon e estar logado na aplicação Amazon GO. A aplicação Amazon GO deverá ser acessada através do CPF cadastrado, gerando uma chave única de identificação e permitir o acesso ao cesto de compras, ao histórico de compras do usuário cadastrado e a geração aleatória de QR-Codes de acesso as filiais Amazon GO. Para que seja possível a geração de um QR-Code de acesso, cada QR-Code identifica uma única conta, porém o sistema deverá reconhecer compras múltiplas de clientes sob o mesmo QR-Code. Cada aplicativo deverá conter apenas um cesto de compras e cada cesto de compras poderá receber diversos produtos seja eles iguais ou diferentes e um mesmo tipo de produto poderá estar presentes em múltiplos cestos de compras. O cliente deverá apresentar o QR-Code gerado pela aplicação em catracas automática na entrada da loja, um moderno sistema de inteligência artificial e visão computação deverá marcar o usuário ou seus dependentes de modo que todos os produtos retirados das prateleiras sejam automaticamente inseridos no cesto de compras do cliente. A desistência da compra e posterior devolução do produto a prateleira deverá acarretar na exclusão do item do cesto de compras. Ao fim das compras, o cliente deverá passar por outro sistema de catracas que fará o desconto da compra em um ou mais cartões de créditos presentes no cadastro. Após o pagamento, o sistema deverá enviar ao cliente um resumo de valor gasto e produtos comprados, assim como essa informação acrescida da data e hora de entrada na filial deverá constar no histórico de compras, acessáveis pela aplicação Amazon GO.

2.1 Requisitos Funcionais

Diferentes grupos de usuários demandarão diferentes operações de manipulação de dados sobre diferentes porções do banco de dados. O grupo AMAZON demandará atualização e recuperação de dados sobre praticamente todas os elementos do banco de dados, uma vez que esse grupo será o responsável por manter os dados atualizados, dando suporte aos outros grupos. O grupo CLIENTE demandará consultas para atualização de seus dados cadastrais e para manipulação de dados sobre seu histórico de compras. O grupo ESTOQUE demandará atualização e recuperação de dados sobre produtos presentes no estoque, sua quantidade, lotes de fabricação e data de validade. O grupo FUNCIONÁRIOS demandará consultas de recuperação de dados sobre suas alocações, informações trabalhistas. O grupo GERAL demandará recuperação de dados sobre valores, localização de lojas, produtos em estoque, horários de funcionamento. O grupo GERÊNCIA demandará a recuperação de dados agrupados sobre funcionários e estoque, tais como carga horária de um funcionário, quantidade de produtos no estoque, quantidade de produtos nas gondolas.

A Tabela 1 apresenta as principais consultas que cada grupo de usuários demandará ao sistema de banco de dados, bem como a frequência esperada de submissão (A para alta, M para média e B para baixa).

Tabela 1: Frequência esperada de consultas por grupo de usuário

Consulta	Grupo	Frequência
Q001 Visualizar quais são suas franquias ativas	AMAZON	M
Q002 Visualizar total de funcionários ativos	AMAZON	B
Q003 Visualizar relatórios de lucros e perdas por filial	AMAZON	A
Q004 Efetivar cadastro de novas filiais	AMAZON	M
Q005 Remover cadastro de filiais	AMAZON	M
Q006 Atualizar dados cadastrais	CLIENTE	B
Q007 Visualizar histórico de compras	CLIENTE	B
Q008 Visualizar compras favoritas	CLIENTE	B
Q009 Recuperar média de gastos	CLIENTE	M
Q010 Visualizar produtos disponíveis no estoque	ESTOQUE	A
Q011 Recuperar a quantidade de cada produto disponível no estoque	ESTOQUE	A
Q012 Visualizar produtos com lote de fabricação específico	ESTOQUE	B
Q013 Recuperar produtos com lote de fabricação específico	ESTOQUE	B
Q014 Visualizar data de validade dos produtos	ESTOQUE	A
Q015 Recuperar produtos com data de validade vencida	ESTOQUE	M
Q016 Recuperar produtos com data de validade próxima ao vencimento	ESTOQUE	M
Q017 Visualizar jornada trabalhada	FUNCIONÁRIO	M
Q018 Recuperar dados de entrada e saída de expediente	FUNCIONÁRIO	B
Q019 Atualizar informações cadastrais da sua função	FUNCIONÁRIO	B
Q020 Visualizar preços de produtos	GERAL	A
Q021 Visualizar endereço de uma filial	GERAL	M
Q022 Visualizar disponibilidade de um produto na loja	GERAL	M
Q023 Visualizar horário de funcionamento da loja	GERAL	B
Q024 Efetivar cadastro de funcionários	GERÊNCIA	M
Q025 Remover funcionário de cadastro	GERÊNCIA	B
Q026 Visualizar carga horária de cada funcionário	GERÊNCIA	M
Q027 Visualizar relatório de estoque	GERÊNCIA	M
Q028 Gerar relatório de lucros e perdas	GERÊNCIA	M
Q029 Gerar relatório de lucros e perdas	GERÊNCIA	M

3 Projeto Conceitual

Essa seção apresenta o projeto conceitual da Amazon GO (v2.0), descrevendo as principais estruturas e restrições conceituais do banco de dados. Particularmente, a Figura 1 apresenta o diagrama entidade-relacionamento (ER) do modelo conceitual do empreendimento.

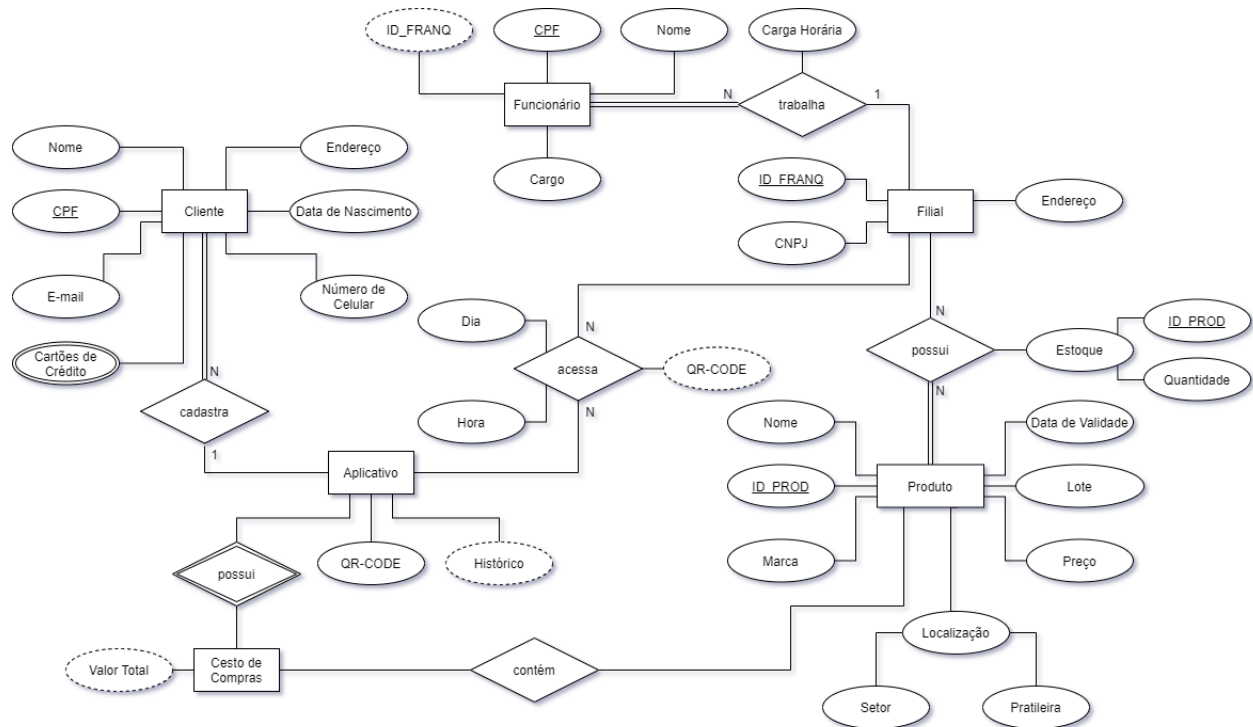


Figura 1: EER do modelo de implementação relacional da Amazon GO (v2.0)

Adicionalmente, a Tabela 2 apresenta com mais detalhes os elementos descritos no diagrama apresentados na Figura 1. Na Tabela 2, podemos observar que foram identificadas na descrição textual do minimundo seis entidades, com uma média de seis atributos por entidade. Além disso, foram identificados sete relacionamentos entre entidades, bem como suas respectivas restrições de cardinalidade, e quatro restrições de totalidade presentes em quatro relacionamentos diferentes.

Tabela 2: Elementos do modelo conceitual do banco de dados Amazon GO (v2.0)

Tipo	Subtipo	ID	Rótulo	Referência	Descrição
Entidade	Forte	E001	Aplicativo		
Entidade	Forte	E002	Cesto de Com- pras		
Entidade	Forte	E003	Cliente		
Entidade	Forte	E004	Funcionário		
Entidade	Forte	E005	Filial		
Entidade	Forte	E006	Produto		
Relacionamento	Forte	R001	cadastra	E001, E003	
Relacionamento	Fraco	R002	possui	E001, E002	
Relacionamento	Forte	R003	acessa	E001, E005	
Relacionamento	Forte	R004	trabalha	E004, E005	
Relacionamento	Forte	R005	possui	E005, E006	
Relacionamento	Forte	R006	contém	E002, E006	
Atributo	Simple	A001	QR-CODE	E001	
Atributo	Derivado	A002	Histórico	E001	
Atributo	Chave	A003	ID _C ESTO	E002	
Atributo	Derivado	A004	Valor Total	E002	
Atributo	Chave	A005	CPF	E003	
Atributo	Simple	A006	Nome	E003	
Atributo	Simple	A007	E-mail	E003	
Atributo	Simple	A008	Endereço	E003	
Atributo	Simple	A009	Data de Nasci- mento	E003	
Atributo	Simple	A010	Número de Celu- lar	E003	
Atributo	Multivalorado	A011	Cartões de Cré- dito	E003	
Atributo	Chave	A012	CPF	E004	
Atributo	Simple	A013	Nome	E004	
Atributo	Simple	A014	Cargo	E004	
Atributo	Derivado	A015	ID _F RANQ	E004	
Atributo	Chave	A016	ID _F RANQ	E005	
Atributo	Simple	A017	CNPJ	E005	
Atributo	Simple	A018	Endereço	E005	
Atributo	Chave	A019	ID _P ROD	E006	
Atributo	Simple	A020	Nome	E006	
Atributo	Simple	A021	Marca	E006	
Atributo	Simple	A022	Data de Validade	E006	
Atributo	Simple	A023	Lote	E006	
Atributo	Simple	A024	Preço	E006	
Atributo	Composto	A025	Localização	E006	
Atributo	Simple	A026	Setor	E006, A025	
Atributo	Simple	A027	Prateleira	E006, A025	
Restrição	Totalidade	C001		R001	E003 total em R001
Restrição	Totalidade	C002		R004	E004 total em R004
Restrição	Totalidade	C003		R005	E006 total em R005
Restrição	Cardinalidade	C004	1-N	R001	E001 função <i>cadastrante</i> (1), E003 função <i>cadastrado</i> (N)
Restrição	Cardinalidade	C005	N-N	R003	E001 função <i>acessante</i> (N), E005 função <i>acessado</i> (N)
Restrição	Cardinalidade	C006	1-N	R004	E005 função <i>contratante</i> (1), E004 função <i>contratado</i> (N)
Restrição	Cardinalidade	C007	N-N	5 R005	E005 função <i>possuidor</i> (1), E006 função <i>possuído</i> (N)
Restrição	Cardinalidade	C008	N-N	R005	E006 função <i>possuidor</i> (1), E006 função <i>contido</i> (N)

4 Projeto Lógico

Essa seção apresenta o projeto lógico do banco de dados Amazon GO (v2.0), descrevendo as principais estruturas e restrições lógicas baseadas no modelo de implementação relacional. Particularmente, a Figura 3 apresenta o diagrama relacional do banco de dados, mapeado a partir do modelo conceitual descrito na Seção 3 do presente relatório.

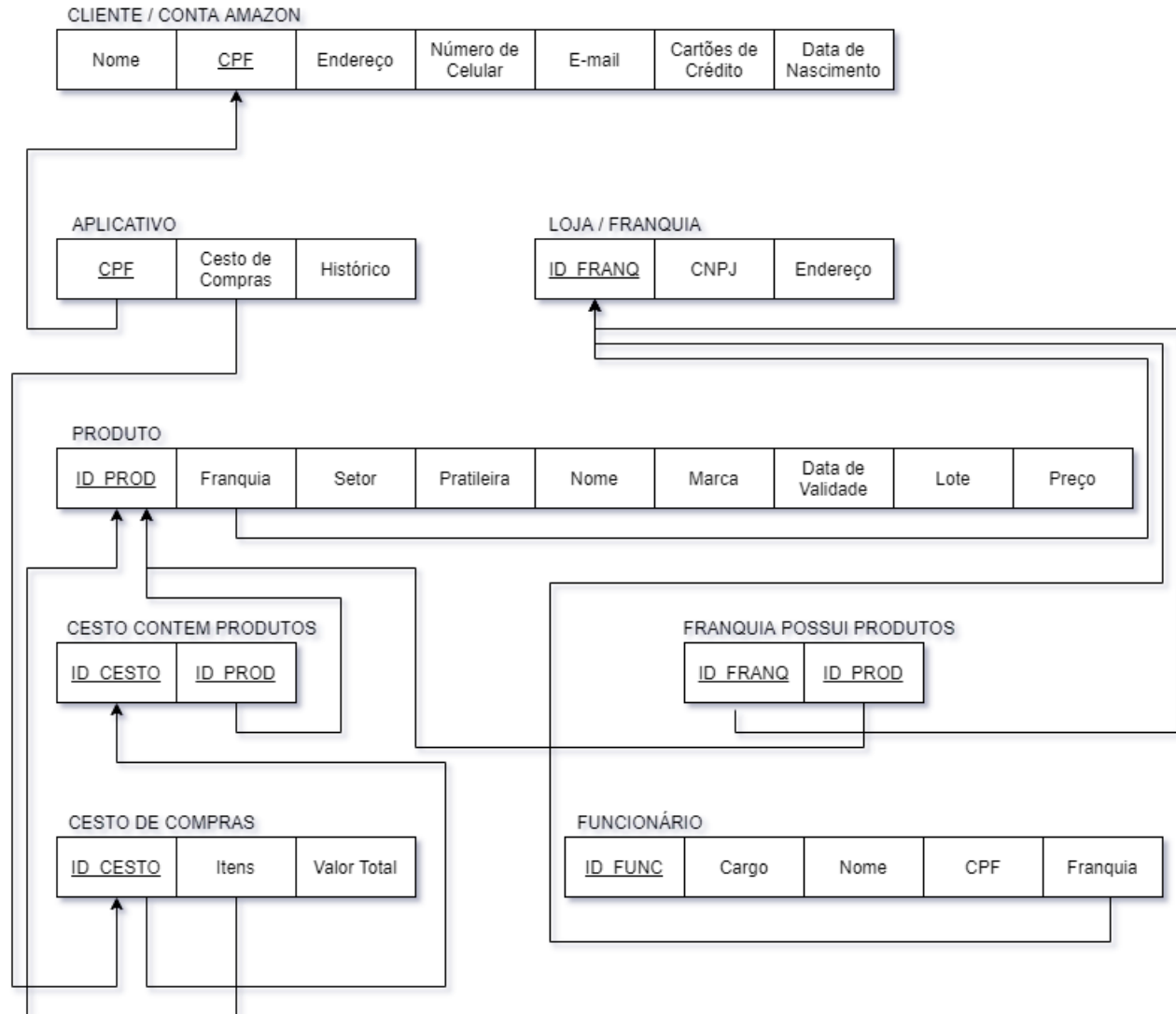


Figura 2: EER do modelo de implementação relacional da Amazon GO (v2.0)

Na Figura 3, podemos observar que foram mapeadas seis relações, sendo que nem todas as relações possuem atributos próprios. Além disso, foram identificados oito referências entre relações. O diagrama relacional apresentado na Figura 2 é muito útil para visualizar de maneira simples e compacta as relações, atributos e restrições de chave e integridade referencial presentes no banco de dados, independentemente do SGBD relacional comercial a ser adotado para sua implementação. O detalhamento de outros tipos de estruturas e restrições sobre dados são dependentes do SGBD relacional comercial a ser adotado.

Além do modelo de dados, uma importante decisão a ser tomada no projeto lógico de um banco de dados é a escolha da abordagem e da solução de banco de dados a serem adotadas. Particularmente, para implementação do Amazon GO (v2.0) adotaremos a abordagem baseada em SGBD relacional e a solução comercial MySQL. A Figura 2 apresenta o EER do modelo de implementação relacional do Amazon GO (v2.0), incluindo restrições de chave, repre-

sentadas como uma figura amarela de chave ao lado esquerdo do rótulo do atributo, tipo, apresentada ao lado direito do rótulo do atributo, nulidade, representada como um losango ao lado esquerdo do rótulo do atributo (losango branco para NULL e azul para NOT NULL), e integridade referencial, com losango vermelho representado chaves estrangeiras.

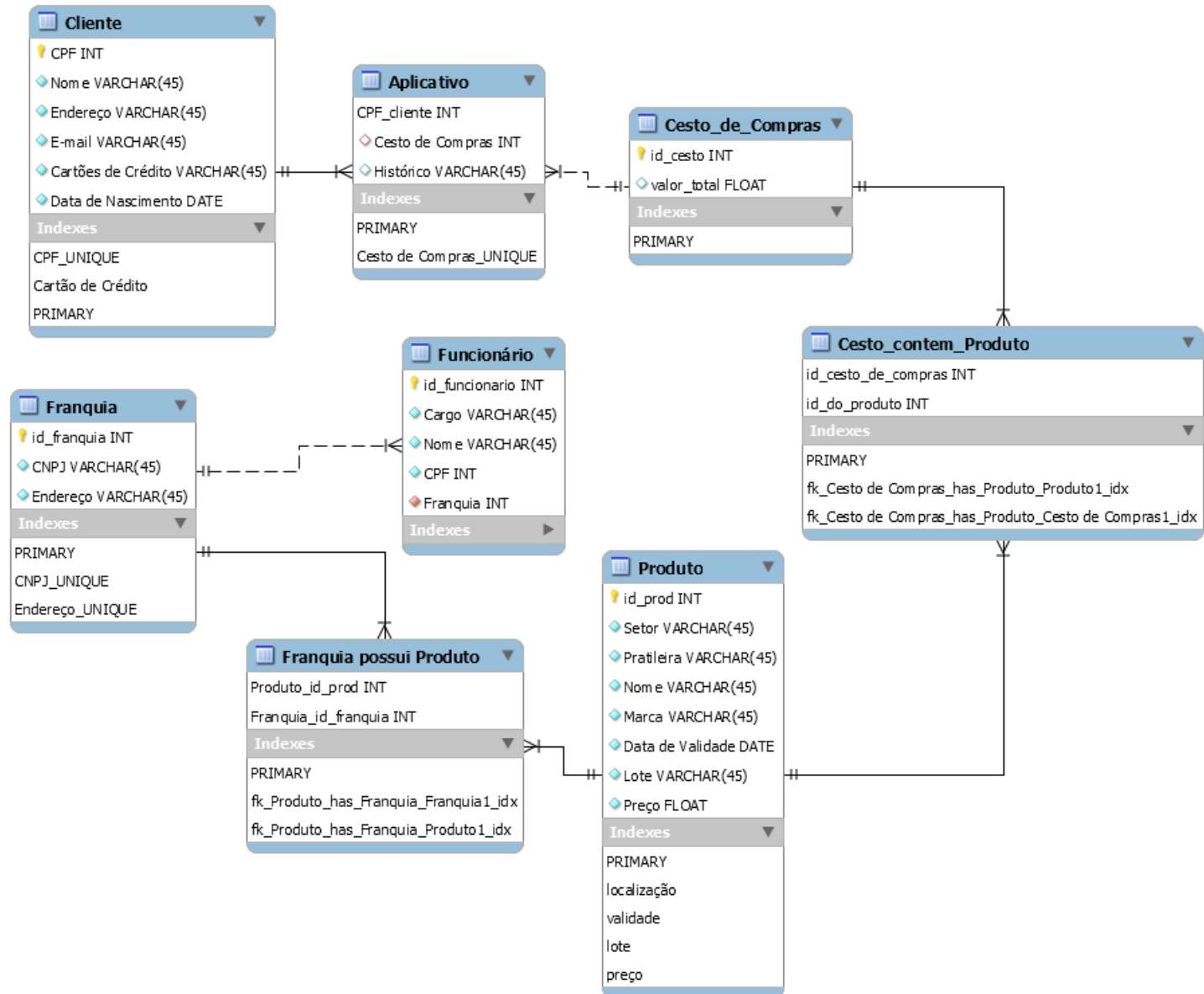


Figura 3: EER do modelo de implementação relacional da Amazon GO (v2.0)

5 Conclusão

O presente relatório apresentou o projeto do banco de dados Amazon GO (v2.0) para um sistema varejista concebido pela Amazon que, em sua versão 2.0, pode ser utilizado por diversas cadeias de lojas que busquem replicar a ideia inovadora de sua criadora. Especificamente, propusemos uma especificação de minimundo e apresentamos os requisitos funcionais e operacionais, o projeto conceitual, lógico e físico do banco de dados, concebido no paradigma relacional e projetado para ser implementado em um SGBD relacional comercial.

6 Anexos: Scripts de Banco de Dados

Scripts de Criação de Tabelas

```
-- MySQL Script generated by MySQL Workbench
-- Tue Dec 1 21:05:53 2020
-- Model: New Model      Version: 2.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES, '
    'NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

DROP SCHEMA IF EXISTS `amazon_go`;

CREATE SCHEMA IF NOT EXISTS `amazon_go` DEFAULT CHARACTER SET utf8 ;

USE `amazon_go` ;

DROP TABLE IF EXISTS `amazon_go`.`Aplicativo` ;
DROP TABLE IF EXISTS `amazon_go`.`Cesto_de_Compras` ;
DROP TABLE IF EXISTS `amazon_go`.`Cliente` ;
DROP TABLE IF EXISTS `amazon_go`.`Franquia` ;
DROP TABLE IF EXISTS `amazon_go`.`Funcionário` ;
DROP TABLE IF EXISTS `amazon_go`.`Produto` ;
DROP TABLE IF EXISTS `amazon_go`.`Cesto_contem_Produto` ;
DROP TABLE IF EXISTS `amazon_go`.`Franquia possui Produto` ;

CREATE TABLE IF NOT EXISTS `amazon_go`.`Cliente` (
    `CPF` INT NOT NULL,
    `Nome` VARCHAR(45) NOT NULL,
    `Endereço` VARCHAR(45) NOT NULL,
    `E-mail` VARCHAR(45) NOT NULL,
    `Cartões de Crédito` VARCHAR(45) NOT NULL,
    `Data de Nascimento` DATE NOT NULL)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `amazon_go`.`Cesto_de_Compras` (
    `id_cesto` INT NOT NULL,
    `valor_total` FLOAT ZEROFILL UNSIGNED NULL)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `amazon_go`.`Aplicativo` (
    `CPF_cliente` INT NOT NULL,
    `Cesto de Compras` INT NULL,
    `Histórico` VARCHAR(45) NULL)
ENGINE= InnoDB;

CREATE TABLE IF NOT EXISTS `amazon_go`.`Franquia` (
    `id_franquia` INT NOT NULL,
    `CNPJ` VARCHAR(45) NOT NULL,
    `Endereço` VARCHAR(45) NOT NULL)
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `amazon_go`.`Funcionário` (
  `id_funcionario` INT NOT NULL,
  `Cargo` VARCHAR(45) NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `CPF` INT UNSIGNED NOT NULL,
  `Franquia` INT NOT NULL)
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `amazon_go`.`Produto` (
  `id_prod` INT NOT NULL,
  `Setor` VARCHAR(45) NOT NULL,
  `prateleira` VARCHAR(45) NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Marca` VARCHAR(45) NOT NULL,
  `Data de Validade` DATE NOT NULL,
  `Lote` VARCHAR(45) NOT NULL,
  `Preço` FLOAT ZEROFILL UNSIGNED NOT NULL)
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `amazon_go`.`Cesto_contem_Produto` (
  `id_cesto_de_compras` INT NOT NULL,
  `id_do_produto` INT NOT NULL)
ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `amazon_go`.`Franquia possui Produto` (
  `Produto_id_prod` INT NOT NULL,
  `Franquia_id_franquia` INT NOT NULL)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Scripts de Criação de Restrições

```
-- MySQL Script generated by MySQL Workbench
-- Tue Dec 1 22:25:37 2020
-- Model: New Model Version: 2.0
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES, '
'NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
USE `amazon_go` ;
```

```
ALTER TABLE `amazon_go`.`cliente`
  ADD CONSTRAINT `PK_Cliente`
  PRIMARY KEY (`CPF`);
```

```

ALTER TABLE `amazon_go`.`cesto_de_compras`
  ADD CONSTRAINT `PK_Cesto`
  PRIMARY KEY (`id_cesto`);

ALTER TABLE `amazon_go`.`aplicativo`
  ADD CONSTRAINT `PK_aplicativo`
  PRIMARY KEY (`CPF_cliente`);

ALTER TABLE `amazon_go`.`produto`
  ADD CONSTRAINT `PK_produto`
  PRIMARY KEY (`id_prod`);

ALTER TABLE `amazon_go`.`franquia`
  ADD CONSTRAINT `PK_franquia`
  PRIMARY KEY (`id_franquia`);

ALTER TABLE `amazon_go`.`funcionário`
  ADD CONSTRAINT `PK_funcionario`
  PRIMARY KEY (`id_funcionario`);

ALTER TABLE `amazon_go`.`cesto_contem_produto`
  ADD CONSTRAINT `PK_cesto_contem_produto`
  PRIMARY KEY (`id_cesto_de_compras`, `id_do_produto`);

ALTER TABLE `amazon_go`.`franquia possui produto`
  ADD CONSTRAINT `PK_franquia_possui_produto`
  PRIMARY KEY (`Produto_id_prod`, `Franquia_id_franquia`);

ALTER TABLE `amazon_go`.`aplicativo`
  ADD CONSTRAINT `FK_CPF_Cliente`
  FOREIGN KEY (`CPF_cliente`)
  REFERENCES `amazon_go`.`Cliente` (`CPF`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `amazon_go`.`aplicativo`
  ADD CONSTRAINT `FK_Cesto`
  FOREIGN KEY (`Cesto de Compras`)
  REFERENCES `amazon_go`.`Cesto_de_Compras` (`id_cesto`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `amazon_go`.`funcionário`
  ADD CONSTRAINT `FK_funcionario`
  FOREIGN KEY (`Franquia`)
  REFERENCES `amazon_go`.`Franquia` (`id_franquia`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `amazon_go`.`cesto_contem_produto`
  ADD CONSTRAINT `FK_cesto_contem_produto_cesto`
  FOREIGN KEY (`id_cesto_de_compras`)
  REFERENCES `amazon_go`.`Cesto_de_Compras` (`id_cesto`)
  ON DELETE CASCADE

```

```

ON UPDATE CASCADE;

ALTER TABLE `amazon_go`.`cesto_contem_produto`
  ADD CONSTRAINT `FK_cesto_contem_produto_produto`
  FOREIGN KEY (`id_do_produto`)
  REFERENCES `amazon_go`.`Produto` (`id_prod`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `amazon_go`.`franquia_possui_produto`
  ADD CONSTRAINT `FK_franquia_possui_produto_produto`
  FOREIGN KEY (`Produto_id_prod`)
  REFERENCES `amazon_go`.`Produto` (`id_prod`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `amazon_go`.`franquia_possui_produto`
  ADD CONSTRAINT `FK_franquia_possui_produto_franquia`
  FOREIGN KEY (`Franquia_id_franquia`)
  REFERENCES `amazon_go`.`Franquia` (`id_franquia`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Scripts de Criação de Restrições (Gatilhos)

Como mencionado no início desta obra, não foram delimitadas nenhuma restrição do tipo trigger/gatilho. Sendo esta restrição necessária para versões futuras deste projeto.

Scripts de Criação de Índices

```

-- MySQL Script generated by MySQL Workbench
-- Tue Dec 1 22:13:42 2020
-- Model: New Model    Version: 2.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES, '
'NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

USE `amazon_go` ;

CREATE UNIQUE INDEX `CPF_UNIQUE`
  ON `amazon_go`.`Cliente` (`CPF` ASC) VISIBLE;

CREATE INDEX `Cartão de Crédito`
  ON `amazon_go`.`Cliente` (`Cartões de Crédito` ASC) VISIBLE;

```

```

CREATE INDEX `Cesto de Compras_UNIQUE`
  ON `amazon_go`.`Aplicativo` (`Cesto de Compras` ASC) INVISIBLE;

CREATE INDEX `localização`
  ON `amazon_go`.`Produto` (`Setor` ASC, `prateleira` ASC) INVISIBLE;

CREATE INDEX `validade`
  ON `amazon_go`.`Produto` (`Data de Validade` ASC) INVISIBLE;

CREATE INDEX `lote`
  ON `amazon_go`.`Produto` (`Lote` ASC) VISIBLE;

CREATE INDEX `preço`
  ON `amazon_go`.`Produto` (`Preço` ASC) VISIBLE;

CREATE UNIQUE INDEX `CNPJ_UNIQUE`
  ON `amazon_go`.`Franquia` (`CNPJ` ASC) VISIBLE;
CREATE UNIQUE INDEX `Endereço_UNIQUE`
  ON `amazon_go`.`Franquia` (`Endereço` ASC) VISIBLE;

CREATE UNIQUE INDEX `CPF_UNIQUE`
  ON `amazon_go`.`Funcionário` (`CPF` ASC) VISIBLE;

CREATE INDEX `FK_franquia_idx`
  ON `amazon_go`.`Funcionário` (`Franquia` ASC) VISIBLE;

CREATE INDEX `fk_Cesto de Compras_has_Produto_Produto1_idx`
  ON `amazon_go`.`Cesto_contem_Produto` (`id_do_produto` ASC) VISIBLE;

CREATE INDEX `fk_Cesto de Compras_has_Produto_Cesto de Compras1_idx`
  ON `amazon_go`.`Cesto_contem_Produto` (`id_cesto_de_compras` ASC) VISIBLE;

CREATE INDEX `fk_Produto_has_Franquia_Franquia1_idx`
  ON `amazon_go`.`Franquia possui Produto` (`Franquia_id_franquia` ASC) VISIBLE;
CREATE INDEX `fk_Produto_has_Franquia_Produto1_idx`
  ON `amazon_go`.`Franquia possui Produto` (`Produto_id_prod` ASC) VISIBLE;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Scripts de Criação do Banco de Dados

```

-- MySQL Script generated by MySQL Workbench
-- Tue Dec 1 22:24:52 2020
-- Model: New Model    Version: 2.0
-- MySQL Workbench Forward Engineering

```

```

START TRANSACTION;
USE `amazon_go`;

```

```

INSERT INTO `amazon_go`.`Cliente`

```

```

(`CPF`, `Nome`, `Endereço`, `E-mail`, `Cartões de Crédito`, `Data de Nascimento`)
VALUES (45486, 'Davi Rodrigues Azevedo', 'Rua João Dalmolin, 1095',
        'dazevedo@jourrapide.com', '5531 0048 0465 1544', '1988-12-12');

INSERT INTO `amazon_go`.`Cliente`
(`CPF`, `Nome`, `Endereço`, `E-mail`, `Cartões de Crédito`, `Data de Nascimento`)
VALUES (65300, 'Maria Cardoso Oliveira', 'Rua Geraldo Alves dos Santos, 822',
        'mariacardoso@dayrep.com', '4539 3432 7782 0268', '1990-05-25');

INSERT INTO `amazon_go`.`Cesto_de_Compras` (`id_cesto`, `valor_total`)
VALUES (0001, NULL);

INSERT INTO `amazon_go`.`Aplicativo` (`CPF_cliente`, `Cesto de Compras`, `Histórico`)
VALUES (45486, 0001, NULL);

INSERT INTO `amazon_go`.`Produto`
(`id_prod`, `Setor`, `Prateleira`, `Nome`, `Marca`, `Data de Validade`,
`Lote`, `Preço`) VALUES
(0001, 'Laticínios', '45A', 'Iogurte Grego', 'Vigor',
'2020-01-12', '456482A', 4.80);

INSERT INTO `amazon_go`.`Produto`
(`id_prod`, `Setor`, `Prateleira`, `Nome`, `Marca`, `Data de Validade`,
`Lote`, `Preço`) VALUES
(0457, 'Alcoólicos', '01X', 'Vinho Carbenet Sauvignon',
'El Toro', '2025-06-25', 'X448AB45', 75.90);

INSERT INTO `amazon_go`.`Franquia` (`id_franquia`, `CNPJ`, `Endereço`)
VALUES (001, '15.436.940/0001-03', '2131 7th Ave, Seattle, WA');

INSERT INTO `amazon_go`.`Funcionário` (`id_funcionario`, `Cargo`, `Nome`, `CPF`,
`Franquia`) VALUES
(045, 'Repositor de Estoque', 'Erika A. Green', 45655, 001);

INSERT INTO `amazon_go`.`Cesto_contem_Produto`
(`id_cesto_de_compras`, `id_do_produto`) VALUES (0001, 0457);

INSERT INTO `amazon_go`.`Franquia possui Produto`
(`Produto_id_prod`, `Franquia_id_franquia`) VALUES (0457, 001);

```